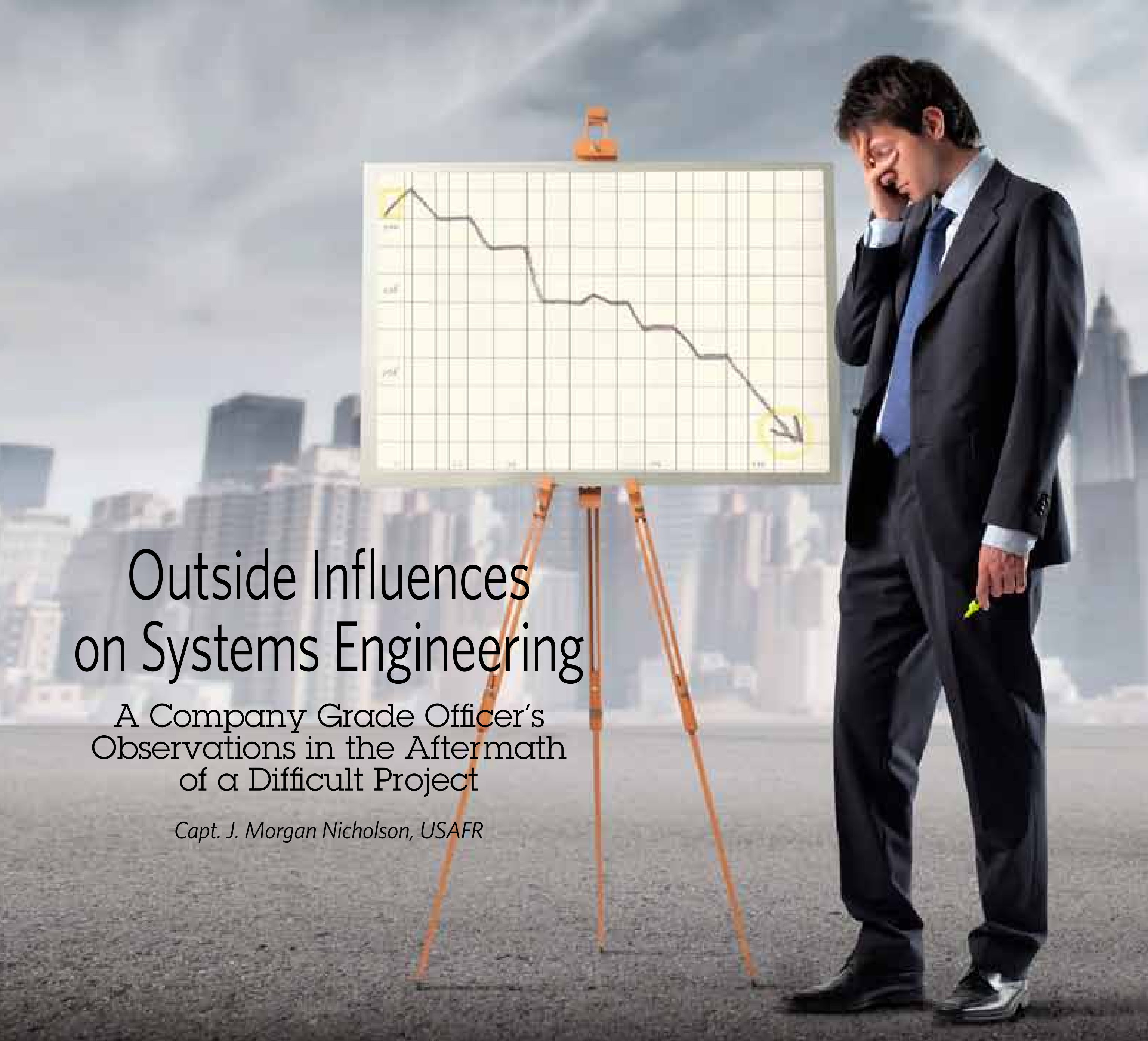

Outside Influences on Systems Engineering

A Company Grade Officer's Observations in the Aftermath of a Difficult Project

Capt. J. Morgan Nicholson, USAFR

In the world of acquisitions management, the systems engineering discipline is often thought of as a *separable, independent* activity that follows a certain flow chart and, if executed correctly, produces a useable item that meets the technical requirements within cost and schedule constraints. This fallacy has no doubt led to many project failures, including the case study presented here. To make matters worse, decisions made in areas thought to be outside of systems engineering are often the root cause of a project's failure. These non-technical decisions have a direct effect on the project's technical performance.

The relationship between engineering and management decisions was once well known, and bridging this gap is one of the reasons that the systems engineering profession came into existence. Unfortunately, this relationship is all too often overlooked, and systems engineering is thought to occur in isolation from management, contracting, logistics, and operations. This attitude

Nicholson *is an Air Force Reservist, an employee of a defense contractor, and a graduate student—all in the field of systems engineering management.*

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **AUG 2011**	2. REPORT TYPE	3. DATES COVERED **00-00-2011 to 00-00-2011**

4. TITLE AND SUBTITLE **Outside Influences on Systems Engineering: A Company Grade Officer's Observations in the Aftermath of a Difficult Project**	5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Defense Acquisition University,Defense AT&L,9820 Belvoir Road,Fort Belvoir,VA,22060-5565**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S) 11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**4**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

can cause many headaches for a project team—and can lead to a project's demise.

A recent project I was part of experienced a series of systems engineering failures, causing the project budget to run over by roughly 300 percent and causing the delivery to take twice as long as anticipated. I inherited this project as lead systems engineer well after the original completion date and well after the system was designed.

Part of my job as lead systems engineer was to determine the causes of the systems engineering failures so they could be prevented in future projects. However, my findings attributed much of the hardship to failures outside the so-called systems engineering process. These failures may have manifested themselves as systems engineering issues, but I believe they were the result of decisions made very early in the project—in some cases, before the project even began. These business decisions rippled through the project unnoticed until project delivery, where they reared their ugly heads and the project spiraled out of control.

Two such decisions had the most severe consequences on the project's outcome. Both were made prior to the project's existence and manifested themselves as systems engineering failures at the end of the project. These insidious failures hid themselves throughout the project and could not be spotted using earned-value management (EVM) techniques or the most elaborate performance metric scheme. Decisions made this early can influence the design of the EVM and performance metrics, making them unable to reveal problems. Worse, these decisions can give the project team false sense of comfort about how the project is progressing.

The project was part of the portfolio of a much larger program providing sustainment and modernization to a number of Air Force weapon systems that are unique but interrelated. This project was a major communication system upgrade to a one-of-a-kind system. The entire portfolio was managed by a single Air Force organization and executed via a long-term, sole-sourced contract. A single contractor was used to execute a number of projects simultaneously across a number of different weapon systems under the umbrella of this single, overarching contract. The contractor was organized into separate product lines, each responsible for the projects associated with a single weapon system. The communication upgrade project was one of the largest (in terms of dollars) and most complex projects attempted on this contract at the time, and therefore drew significant attention from our leadership.

The Contract Structure

The contract type used for this project was a cost-plus-award-fee contract. This means that the government paid all project costs incurred by the contractor and paid the contractor's profit based on an award-fee plan. In essence, the government assumed all the risk; if the contractor did not deliver, the government gave it more money to complete the project. All the contractor risked was the award fee. This is different from a firm-fixed-price contract, in which the contractor is required to finish the project without additional cost to the government in the case of an overrun.

A cost-plus contract may indeed be the appropriate contracting strategy for this effort. With this strategy, the award-fee plan is the critical document the government uses to tell the contractor what the award fee (profit) will be based on. In other words, the award fee plan is how the government tells the contractor what is important and what is not. Furthermore, the government can quantify how much more important one deliverable is than another.

This is one area where the government failed on this project. The government did not effectively tell the contractor what it wanted; the government did not communicate that *delivering* the *right* product, *on time* and *on budget* was most important. Instead, the government tried to develop an award-fee plan that distributed profit evenly to the contractor throughout the fiscal year. Although this is good for the contractor, the project kick-off and planning phases had more profit associated with them than developmental test and evaluation (DT&E). As the customer, what would be more important to you—the method and timing that the contractor used to plan the project in the beginning? Or the successful integration, test and formal delivery of the product at the end?

Perhaps the most critical failure was a decision made before the project was even a formal project. The contractor was repeatedly blamed for the 300 percent cost overrun and 200 percent schedule delay. The contractor should still have received the majority of profit for successful project planning, requirements review, design review, documentation delivery, EVM reporting, etc. But the contractor attempted to cover the cost overruns by forfeiting its award fee and using the money to cover these costs—a calculated business decision that made sense in protecting the other projects in its portfolio.

How does this affect systems engineering? If the government tells the contractor that tasks such as EVM reporting are of equal importance to systems integration, the contractor will create, tailor, and follow processes that maximize its profit. The end result is an equal emphasis on EVM reporting and systems integration.

This problem was never noticed during the project. In fact, the majority of the projects on this contract are structured the exact same way. Since the overarching contract is structured this way, the award-fee reporting and EVM systems are also based on this design. Thus a project can appear to be chugging right along with great interim award-fee scores and impeccable EVM numbers—but secretly be heading for a train wreck.

The surprising failure for this project occurred during DT&E, when we found the software wasn't stable. In fact, the software crashed after 40 seconds of being "live" on the system.

This colossal failure was attributed to an incompetent engineering team, and in its aftermath, the project manager and entire engineering team were replaced. However, the award-fee score to this point (including the DT&E failure) was greater than 90 percent, and the EVM metrics were still within acceptable thresholds.

Test should always be an independent entity and should have a separate chain of command.

How can this problem be fixed? The contracting personnel who develop the award-fee plan should consider systems engineering in their planning. The award-fee emphasis should reflect what is most important to the government—successful project delivery. If 90 percent of the award fee (rather than 5 percent) had been based on DT&E, I suspect the contractor would design processes to help ensure successful DT&E completion. After all, does it really matter when the contractor holds kick-off meetings or when design reviews take place if the project is delivered on time nd within budget?

The 'Org Chart'

The paradigm used by the contractor to organize itself also creates challenges for systems engineering. The contractor for this project primarily uses a projectized organizational structure, which offers a number of advantages: strong communication channels, very rapid response time, loyalty to the project, and ability to maintain key expertise.

In theory, a projectized organizational structure makes sense for a product line that consists of a single, one-of-a-kind system. However, expertise becomes very "stovepiped" and is not shared in the organization.

I once worked on a project that involved designing a programmable logic controller (PLC) to manage the cooling air for an electro-optical system. Having spent several years as a control-system technician, I was shocked to find that nobody who engineered the system had any control-system expertise; this is a highly specialized field, and these tasks are typically accomplished by highly specialized personnel.

Not surprisingly, this project had a number of problems in quality audits, testing, and integration. The organization employed a number of engineers with control-system expertise, but they were allocated to a different project on a different product line, so these resources were not shared.

The communication-upgrade project had a similar problem. The project involved not only communications engineering (a highly specialized field) but also a highly irregular, specialized type of communication protocol. The project team did not have any communication experts. Furthermore, they did not employ anyone with knowledge of this protocol, intending instead to "build expertise inside the product line." They did this despite having communications engineers in the organization from other product lines and despite the government's request for them to leverage this expertise.

Note that I said the contractor *primarily* uses a projectized organizational structure. Some personnel are occasionally matrixed to the product line for functions such as systems engineering, logistics, drafting, and configuration management.

Did you notice that "test" was not in the list? This is because the so-called "independent test team" reports directly to the product line manager. This is a fundamental flaw in this organization structure. Test should always be an independent entity and should have a separate chain of command. For example, the Air Force Operational Test and Evaluation Center (AFOTEC) reports directly to the Headquarters Air Force rather than a major Air Force command, such as Space Command or Materiel Command. This ensures the requirements are being independently verified and helps reduce the influence of cost and schedule pressures.

For this project, lack of test independence was often a problem. The contractor's product line manager often agreed to ridiculous test deadlines and objectives despite the objections of his test lead. On one occasion, the test lead actually had to leave the meeting because she was so upset with the product line manager. Moreover, the test lead was actually "shushed" in a technical meeting when she tried to report that a particular requirement was not being met.

This lack of test independence led the project down a number of paths that were to its detriment. Many times, the software was thought to be ready for release only to find critical defects during government acceptance testing. These defects caused serious cost and schedule impacts that could have been avoided—not to mention the failures in customer-expectation management.

The contractor also had a separate functional division inappropriately named "systems engineering." This division typically contained the "best and brightest" engineers in the organization, with a comprehensive understanding of the systems in the portfolio. These engineers were often "promoted" from the product lines to the systems engineering division and focused primarily on advanced concepts and big-picture kinds of issues.

The project had a number of critical defects that tied directly to incorrect requirements. The project ran over by roughly 300 percent, and three-quarters of the overrun costs were devoted

to fixing defects—many of which could be traced *directly* to an incorrect requirement. Most of these incorrect requirements could have been prevented by including more system experts in requirement development. These system experts were not available to the project team because they were part of the systems engineering division and because of the project-based nature of the organization. The flaw was not in the engineering process itself but in its execution, due to a lack of expertise.

The organizational structure used for the project set the stage for a number of problems to manifest themselves during integration and testing—particularly a lack of system expertise and independent test activities. Once again, the failures appeared on the surface as engineering failures, such as poor programming and poor unit testing. However, poor programming and poor testing were a result of poor systems engineering and a lack of test team independence—both of which originated with the org chart.

Conclusion

Many systems engineering problems in the real world are more than just process gaps in systems engineering; they are often symptoms of business decisions that *manifest* themselves in systems engineering. A poor organizational structure creates a lack of systems engineering expertise, which leads to poor requirements specifications. This is manifested as a series of critical defects during formal DT&E. A poor contracting strategy sets the stage for a systems engineering strategy that focuses on following the process rather than delivering a successful product, on time and within budget.

You might be thinking, "Well, this is obvious." But it is rarely addressed in any systems engineering textbook or graduate course. Systems engineering is treated as an independent, objective entity that directs the development effort, with the end user fully considered and acting as an advocate of the customer and a check-and-balance between the business and technical aspects of the project.

In reality, these functions are so closely coupled that they should not be thought of as independent at all. Management questions such as "How do we organize ourselves to minimize overhead?" should not be answered without considering the impact on the end product. Moving all the systems experts out of the divisions that work on the systems doesn't make sense from a technical perspective. However, from a business perspective, it minimizes overhead and makes a nice-looking org chart.

It is often stated that systems engineering processes should be applied throughout the project life cycle. This is true. But what about prior to the project? Does "cradle to grave" really encompass everything? Can a project be doomed before the need is even conceived? Perhaps it can, and systems engineering should be a serious topic of discussion when the organization is formed or when the contracting strategy is outlined.

The author can be contacted at **j.nicholson.ctr@smdck.smdc.army.mil.**